

AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Previously Presented) A computer-readable medium encoded with a data structure, the data structure for a type system implemented within a computing environment and the data structure providing requested services on an artifact in the type system, the data structure comprising:
 - a) a ClrElement base class for capturing common functionality of objects of the type system, the ClrElement base class comprising data members AttributeDeclaration, DocSummary, DocRemarks, IsEditable, IsInjected, IsCodeParseable, and IsFromReferenceAssemblies;
 - b) at least one controller object, the controller object in communication with the base class, the at least one controller object validating the requested services based on a set of rules associated with a programming language; and
 - c) a first class providing a level of abstraction between a second class and a third class, the second class and the third class searchable by the first class.
2. (Original) The computer-readable medium of claim 1, wherein the artifact comprises one of a namespace, a class, an interface, an enumeration, a delegate, an attribute, a field, a property, and an event.
3. (Original) The computer-readable medium of claim 1, wherein the programming language comprises one of Visual Basic, C++, C#, and J#.
4. (Original) The computer-readable medium of claim 1, wherein the base class determines the at least one controller object to communicate with in order to validate the requested services.
5. (Cancelled).

6. (Currently Amended) The computer-readable medium of claim 1~~claim 5~~, wherein the second class and the third class comprise nested classes.
7. (Currently Amended) The computer-readable medium of claim 1~~claim 5~~, wherein the second class and the third class include nested namespaces.
8. (Previously Presented) The computer-readable medium of claim 1, wherein the data structure further comprises:
 - d) a container for storing types in the type system.
9. (Original) The computer-readable medium of claim 1, wherein the requested services comprise modifying the artifact in the type system.
10. (Original) The computer-readable medium of claim 1, wherein the requested services comprise creating a new artifact in the type system.

11. (Previously Presented) A method of modifying an artifact for use in a type system meta-model, the method comprising:

- d) receiving a request from an application programming interface to modify an artifact in the type system meta-model, wherein the type system meta-model comprises a ClrElement base class for capturing common functionality of objects of the type system, the ClrElement base class comprising data members AttributeDeclaration, DocSummary, DocRemarks, IsEditable, IsInjected, IsCodeParseable, and IsFromReferenceAssemblies, and the type system meta-model includes a first class providing a level of abstraction between a second class and a third class, the second class and the third class searchable by the first class;
- e) in response to issuing at least one instruction to a language specific controller object, the language specific controller object validating the request based on rules associated with a programming language; and
- f) in response to a validated request from the language specific controller, modifying the artifact.

12. (Original) The method of modifying an artifact for use in a type system meta-model of claim 11, wherein the method further comprises the step of:

- d) transmitting a response to the application programming interface that the artifact has been modified.

13. (Original) The method of modifying an artifact for use in a type system meta-model of claim 11, wherein the artifact comprises one of a namespace, a class, an interface, an enumeration, a delegate, an attribute, a field, a property, and an event.

14. (Original) The method of modifying an artifact for use in a type system meta-model of claim 11, wherein the programming language comprises one of Visual Basic, C++, C#, and J#.

15. (Previously Presented) A method of creating an artifact for use in a type system meta-model, the method comprising:

- g) receiving a request from an application programming interface to create an artifact in the type system meta-model, wherein the type system meta-model comprises a `ChElement` base class for capturing common functionality of objects of the type system, the `ChElement` base class comprising data members `AttributeDeclaration`, `DocSummary`, `DocRemarks`, `IsEditable`, `IsInjected`, `IsCodeParseable`, and `IsFromReferenceAssemblies`, and the type system meta-model includes a first class providing a level of abstraction between a second class and a third class, the second class and the third class searchable by the first class;
- h) in response to issuing at least one instruction to a language specific controller object, the language specific controller object validating the request based on rules associated with a programming language; and
- i) in response to a validated request from the language specific controller, creating the artifact.

16. (Original) The method of creating an artifact for use in a type system meta-model of claim 15, wherein the method further comprises the step of:

- d) transmitting a response to the application programming interface that the artifact has been created.

17. (Original) The method of creating an artifact for use in a type system meta-model of claim 15, wherein the artifact comprises one of a namespace, a class, an interface, an enumeration, a delegate, an attribute, a field, a property, and an event.

18. (Original) The method of creating an artifact for use in a type system meta-model of claim 15, wherein the programming language comprises one of Visual Basic, C++, C#, and J#.

19. (Previously Presented) The method of modifying an artifact for use in a type system meta-model of claim 11, the second class and the third class comprise nested classes.
20. (Previously Presented) The method of modifying an artifact for use in a type system meta-model of claim 11, the second class and the third class include nested namespaces.
21. (Previously Presented) The method of creating an artifact for use in a type system meta-model of claim 15, the second class and the third class comprise nested classes.
22. (Previously Presented) The method of creating an artifact for use in a type system meta-model of claim 15, the second class and the third class include nested namespaces.